

FILEID**PATFRE

PPPPPPPP PAAAAAA TTTTTTTTTT FFFFFFFF RRRRRRRR EEEEEEEE
PPPPPPPP PAAAAAA TTTTTTTTTT FFFFFFFF RRRRRRRR EEEEEEEE
PP PP AA AA TT FF RR RR EE
PP PP AA AA TT FF RR RR EE
PP PP AA AA TT FF RR RR EE
PPPPPPPP AA AA TT FFFFFF RRRRRRRR EEEEEEEE
PPPPPPPP AA AA TT FFFFFF RRRRRRRR EEEEEEEE
PP AAAAAAAA TT FF RR RR EE
PP AAAAAAAA TT FF RR RR EE
PP AA AA TT FF RR RR EE
PP AA AA TT FF RR RR EE
PP AA AA TT FF RR RR EEEEEEEE
PP AA AA TT FF RR RR EEEEEEEE

....
....
....
....

LL IIIII SSSSSSS
LL IIIII SSSSSSS
LL II SS SSSSSSS
LLLLLLLLL IIIII SSSSSSS
LLLLLLLLL IIIII SSSSSSS

```
1 0001 0 MODULE PATFRE (
2 0002 0     XIF %VARIANT EQL 1
3 0003 0     XTHEN
4 0004 0         ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE),
5 0005 0     XFI
6 0006 0         IDENT = 'V04-000' =
7 0007 1 BEGIN
8 0008 1 !*****
9 0009 1 !*
10 0010 1 !* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 !* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 !* ALL RIGHTS RESERVED.
13 0013 1 !*
14 0014 1 !* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 !* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 !* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 !* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 !* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 !* TRANSFERRED.
20 0020 1 !*
21 0021 1 !* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 !* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 !* CORPORATION.
24 0024 1 !*
25 0025 1 !* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 !* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 !*
28 0028 1 !*
29 0029 1 !*
30 0030 1 !*****
31 0031 1 !*
32 0032 1 FACILITY: PATCH
33 0033 1
34 0034 1 ++
35 0035 1 FUNCTIONAL DESCRIPTION:
36 0036 1
37 0037 1     Free storage allocator and manager for symbol table.
38 0038 1
39 0039 1 Version: V02-008
40 0040 1
41 0041 1 History:
42 0042 1     Author:
43 0043 1     Isaac Nassi, 7 Jul 1976: Version 01
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1     V02-008 PCG0001 Peter George 02-FEB-1981
48 0048 1     Add require statement for LIBS:PATDEF.REQ
49 0049 1
50 0050 1     Modified by:
51 0051 1     Carol Peters, 11 Oct 1977: Version 13
52 0052 1
53 0053 1     Modified by:
54 0054 1     Kathleen Morse, 13 Oct 1977: Version 14
55 0055 1
56 0056 1     Revision history:
57 0057 1
```

	NO	DATE	PROGRAMMER	PURPOSE		
58	0058	1	NO	-----		
59	0059	1	--	-----		
60	0060	1				
61	0061	1	00	13-OCT-77	K.D. MORSE	MODIFY VERSION 13 FOR PATCH.
62	0062	1	01	27-DEC-77	K.D. MORSE	ADD ROUTINE PAT\$REPORT FREE.
63	0063	1	02	27-DEC-77	K.D. MORSE	INITIALIZE PAT\$GL RST BEGN.
64	0064	1	03	5-JAN-78	K.D. MORSE	NO CHANGES FOR VERS 14-16.
65	0065	1	04	14-APR-78	K.D. MORSE	NO CHANGES FOR VERS 17.
66	0066	1	05	25-APR-78	K.D. MORSE	CONVERT TO NATIVE COMPILER.
67	0067	1	06	18-MAY-78	K.D. MORSE	NO CHANGES FOR VERS 18-19.
68	0068	1	07	13-JUN-78	K.D. MORSE	ADD FAO COUNTS TO SIGNALS.
69	0069	1	--			
70	0070	1	--			

: R

```

0071 1 +++
0072 1 Abstract:
0073 1
0074 1 THIS MODULE CONTAINS PROCEDURES TO MANAGE AN AREA OF FREE
0075 1 STORAGE.
0076 1
0077 1 THE ROUTINE PAT$FREEINIT IS CALLED AT SYSTEM INITIALIZATION
0078 1 TO INITIALIZE THE MODULE. IT TAKES AS INPUT THE ADDRESS
0079 1 OF THE PROCEDURE TO BE CALLED IN THE EVENT AN ERROR
0080 1 IS ENCOUNTERED. CODES FOR ERROR CONDITIONS ARE CONTAINED IN
0081 1 FILE 'PATMSG.REQ'.
0082 1
0083 1 THE ROUTINE FREEGET IS CALLED TO ALLOCATE A USER SPECIFIED
0084 1 NUMBER OF LONGWORDS. IF THE REQUEST CAN BE SATISFIED,
0085 1 FREEGET RETURNS A POINTER TO THE BLOCK OF LONGWORDS.
0086 1 THE RETURNED BLOCK CONTAINS ONE EXTRA WORD CONTAINING
0087 1 THE SIZE REQUEST AT WORD -1 IN THE BLOCK. THIS SIZE
0088 1 IS CHECKED WHEN THE STORAGE IS RETURNED. IT SHOULD
0089 1 NOT BE MODIFIED. THE ALGORITHM USED IS A FIRST FIT
0090 1 ALGORITHM WHICH WHILE NOT OPTIMAL SHOULD GIVE REASONABLE
0091 1 RESULTS, AND DO A MINIMUM OF SEARCHING.
0092 1
0093 1 Dynamic storage is the last 64K bytes of the per process
0094 1 address space. It is made accessible by a SCRETVA system
0095 1 service call made in FREEINIT.
0096 1
0097 1 PAT$FREERELEASE IS CALLED TO RELEASE STORAGE NO LONGER NEEDED.
0098 1 IT ATTEMPTS TO DO AS MUCH COMPACTION AS IS POSSIBLE.
0099 1
0100 1 ROUTINE PAT$FREEZ IS CALLED TO ALLOCATE A BLOCK OF
0101 1 CLEARED FREE STORAGE.
0102 1
0103 1 ROUTINE PAT$REPORT_FREE REPORTS THE NUMBER OF BYTES LEFT IN FREE STORAGE.
0104 1
0105 1 --
0106 1
0107 1
0108 1 TABLE OF CONTENTS
0109 1
0110 1
0111 1 FORWARD ROUTINE
0112 1 PAT$FREEINIT : NOVALUE,                                ! routine to initialize free storage
0113 1 FREEGET,                                         ! routine to get some free storage
0114 1 PAT$FREERELEASE : NOVALUE,                           ! routine to release some free storage
0115 1 PAT$FREEZ,                                         ! routine to get and zero some free storage
0116 1 PAT$REPORT_FREE;                                 ! ROUTINE TO REPORT FREE STORAGE LEFT
0117 1
0118 1
0119 1 INCLUDE FILES
0120 1
0121 1
0122 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';
0123 1 REQUIRE 'SRC$:PATPCT.REQ';
0124 1 REQUIRE 'SRC$:VXSMAC.REQ';
0125 1 REQUIRE 'SRC$:PATGEN.REQ';
0126 1 REQUIRE 'SRC$:BSTRUC.REQ';
0127 1 REQUIRE 'LIBS:PATDEF.REQ';
0128 1 REQUIRE 'LIBS:LIBDEF.REQ';
0129 1
0130 1
0131 1
0132 1
0133 1
0134 1
0135 1
0136 1
0137 1
0138 1
0139 1
0140 1
0141 1
0142 1
0143 1
0144 1
0145 1
0146 1
0147 1
0148 1
0149 1
0150 1
0151 1
0152 1
0153 1
0154 1
0155 1
0156 1
0157 1
0158 1
0159 1
0160 1
0161 1
0162 1
0163 1
0164 1
0165 1
0166 1
0167 1
0168 1
0169 1
0170 1
0171 1
0172 1
0173 1
0174 1
0175 1
0176 1
0177 1
0178 1
0179 1
0180 1
0181 1
0182 1
0183 1
0184 1
0185 1
0186 1
0187 1
0188 1
0189 1
0190 1
0191 1
0192 1
0193 1
0194 1
0195 1
0196 1
0197 1
0198 1
0199 1
0200 1
0201 1
0202 1
0203 1
0204 1
0205 1
0206 1
0207 1
0208 1
0209 1
0210 1
0211 1
0212 1
0213 1
0214 1
0215 1
0216 1
0217 1
0218 1
0219 1
0220 1
0221 1
0222 1
0223 1
0224 1
0225 1
0226 1
0227 1
0228 1
0229 1
0230 1
0231 1
0232 1
0233 1
0234 1
0235 1
0236 1
0237 1
0238 1
0239 1
0240 1
0241 1
0242 1
0243 1
0244 1
0245 1
0246 1
0247 1
0248 1
0249 1
0250 1
0251 1
0252 1
0253 1
0254 1
0255 1
0256 1
0257 1
0258 1
0259 1
0260 1
0261 1
0262 1
0263 1
0264 1
0265 1
0266 1
0267 1
0268 1
0269 1
0270 1
0271 1
0272 1
0273 1
0274 1
0275 1
0276 1
0277 1
0278 1
0279 1
0280 1
0281 1
0282 1
0283 1
0284 1
0285 1
0286 1
0287 1
0288 1
0289 1
0290 1
0291 1
0292 1
0293 1
0294 1
0295 1
0296 1
0297 1
0298 1
0299 1
0300 1
0301 1
0302 1
0303 1
0304 1
0305 1
0306 1
0307 1
0308 1
0309 1
0310 1
0311 1
0312 1
0313 1
0314 1
0315 1
0316 1
0317 1
0318 1
0319 1
0320 1
0321 1
0322 1
0323 1
0324 1
0325 1
0326 1
0327 1
0328 1
0329 1
0330 1
0331 1
0332 1
0333 1
0334 1
0335 1
0336 1
0337 1
0338 1
0339 1
0340 1
0341 1
0342 1
0343 1
0344 1
0345 1
0346 1
0347 1
0348 1
0349 1
0350 1
0351 1
0352 1
0353 1
0354 1
0355 1
0356 1
0357 1
0358 1
0359 1
0360 1
0361 1
0362 1
0363 1
0364 1
0365 1
0366 1
0367 1
0368 1
0369 1
0370 1
0371 1
0372 1
0373 1
0374 1
0375 1
0376 1
0377 1
0378 1
0379 1
0380 1
0381 1
0382 1
0383 1
0384 1
0385 1
0386 1
0387 1
0388 1
0389 1
0390 1
0391 1
0392 1
0393 1
0394 1
0395 1
0396 1
0397 1
0398 1
0399 1
0400 1
0401 1
0402 1
0403 1
0404 1
0405 1
0406 1
0407 1
0408 1
0409 1
0410 1
0411 1
0412 1
0413 1
0414 1
0415 1
0416 1
0417 1
0418 1
0419 1
0420 1
0421 1
0422 1
0423 1
0424 1
0425 1
0426 1
0427 1
0428 1
0429 1
0430 1
0431 1
0432 1
0433 1
0434 1
0435 1
0436 1
0437 1
0438 1
0439 1
0440 1
0441 1
0442 1
0443 1
0444 1
0445 1
0446 1
0447 1
0448 1
0449 1
0450 1
0451 1
0452 1
0453 1
0454 1
0455 1
0456 1
0457 1
0458 1
0459 1
0460 1
0461 1
0462 1
0463 1
0464 1
0465 1
0466 1
0467 1
0468 1
0469 1
0470 1
0471 1
0472 1
0473 1
0474 1
0475 1
0476 1
0477 1
0478 1
0479 1
0480 1
0481 1
0482 1
0483 1
0484 1
0485 1
0486 1
0487 1
0488 1
0489 1
0490 1
0491 1
0492 1
0493 1
0494 1
0495 1
0496 1
0497 1
0498 1
0499 1
0500 1
0501 1
0502 1
0503 1
0504 1
0505 1
0506 1
0507 1
0508 1
0509 1
0510 1
0511 1
0512 1
0513 1
0514 1
0515 1
0516 1
0517 1
0518 1
0519 1
0520 1
0521 1
0522 1
0523 1
0524 1
0525 1
0526 1
0527 1
0528 1
0529 1
0530 1
0531 1
0532 1
0533 1
0534 1
0535 1
0536 1
0537 1
0538 1
0539 1
0540 1
0541 1
0542 1
0543 1
0544 1
0545 1
0546 1
0547 1
0548 1
0549 1
0550 1
0551 1
0552 1
0553 1
0554 1
0555 1
0556 1
0557 1
0558 1
0559 1
0560 1
0561 1
0562 1
0563 1
0564 1
0565 1
0566 1
0567 1
0568 1
0569 1
0570 1
0571 1
0572 1
0573 1
0574 1
0575 1
0576 1
0577 1
0578 1
0579 1
0580 1
0581 1
0582 1
0583 1
0584 1
0585 1
0586 1
0587 1
0588 1
0589 1
0590 1
0591 1
0592 1
0593 1
0594 1
0595 1
0596 1
0597 1
0598 1
0599 1
0600 1
0601 1
0602 1
0603 1
0604 1
0605 1
0606 1
0607 1
0608 1
0609 1
0610 1
0611 1
0612 1
0613 1
0614 1
0615 1
0616 1
0617 1
0618 1
0619 1
0620 1
0621 1
0622 1
0623 1
0624 1
0625 1
0626 1
0627 1
0628 1
0629 1
0630 1
0631 1
0632 1
0633 1
0634 1
0635 1
0636 1
0637 1
0638 1
0639 1
0640 1
0641 1
0642 1
0643 1
0644 1
0645 1
0646 1
0647 1
0648 1
0649 1
0650 1
0651 1
0652 1
0653 1
0654 1
0655 1
0656 1
0657 1
0658 1
0659 1
0660 1
0661 1
0662 1
0663 1
0664 1
0665 1
0666 1
0667 1
0668 1
0669 1
0670 1
0671 1
0672 1
0673 1
0674 1
0675 1
0676 1
0677 1
0678 1
0679 1
0680 1
0681 1
0682 1
0683 1
0684 1
0685 1
0686 1
0687 1
0688 1
0689 1
0690 1
0691 1
0692 1
0693 1
0694 1
0695 1
0696 1
0697 1
0698 1
0699 1
0700 1
0701 1
0702 1
0703 1
0704 1
0705 1
0706 1
0707 1
0708 1
0709 1
0710 1
0711 1
0712 1
0713 1
0714 1
0715 1
0716 1
0717 1
0718 1
0719 1
0720 1
0721 1
0722 1
0723 1
0724 1
0725 1
0726 1
0727 1
0728 1
0729 1
0730 1
0731 1
0732 1
0733 1
0734 1
0735 1
0736 1
0737 1
0738 1
0739 1
0740 1
0741 1
0742 1
0743 1
0744 1
0745 1
0746 1
0747 1
0748 1
0749 1
0750 1
0751 1
0752 1
0753 1
0754 1
0755 1
0756 1
0757 1
0758 1
0759 1
0760 1
0761 1
0762 1
0763 1
0764 1
0765 1
0766 1
0767 1
0768 1
0769 1
0770 1
0771 1
0772 1
0773 1
0774 1
0775 1
0776 1
0777 1
0778 1
0779 1
0780 1
0781 1
0782 1
0783 1
0784 1
0785 1
0786 1
0787 1
0788 1
0789 1
0790 1
0791 1
0792 1
0793 1
0794 1
0795 1
0796 1
0797 1
0798 1
0799 1
0800 1
0801 1
0802 1
0803 1
0804 1
0805 1
0806 1
0807 1
0808 1
0809 1
0810 1
0811 1
0812 1
0813 1
0814 1
0815 1
0816 1
0817 1
0818 1
0819 1
0820 1
0821 1
0822 1
0823 1
0824 1
0825 1
0826 1
0827 1
0828 1
0829 1
0830 1
0831 1
0832 1
0833 1
0834 1
0835 1
0836 1
0837 1
0838 1
0839 1
0840 1
0841 1
0842 1
0843 1
0844 1
0845 1
0846 1
0847 1
0848 1
0849 1
0850 1
0851 1
0852 1
0853 1
0854 1
0855 1
0856 1
0857 1
0858 1
0859 1
0860 1
0861 1
0862 1
0863 1
0864 1
0865 1
0866 1
0867 1
0868 1
0869 1
0870 1
0871 1
0872 1
0873 1
0874 1
0875 1
0876 1
0877 1
0878 1
0879 1
0880 1
0881 1
0882 1
0883 1
0884 1
0885 1
0886 1
0887 1
0888 1
0889 1
0890 1
0891 1
0892 1
0893 1
0894 1
0895 1
0896 1
0897 1
0898 1
0899 1
0900 1
0901 1
0902 1
0903 1
0904 1
0905 1
0906 1
0907 1
0908 1
0909 1
0910 1
0911 1
0912 1
0913 1
0914 1
0915 1
0916 1
0917 1
0918 1
0919 1
0920 1
0921 1
0922 1
0923 1
0924 1
0925 1
0926 1
0927 1
0928 1
0929 1
0930 1
0931 1
0932 1
0933 1
0934 1
0935 1
0936 1
0937 1
0938 1
0939 1
0940 1
0941 1
0942 1
0943 1
0944 1
0945 1
0946 1
0947 1
0948 1
0949 1
0950 1
0951 1
0952 1
0953 1
0954 1
0955 1
0956 1
0957 1
0958 1
0959 1
0960 1
0961 1
0962 1
0963 1
0964 1
0965 1
0966 1
0967 1
0968 1
0969 1
0970 1
0971 1
0972 1
0973 1
0974 1
0975 1
0976 1
0977 1
0978 1
0979 1
0980 1
0981 1
0982 1
0983 1
0984 1
0985 1
0986 1
0987 1
0988 1
0989 1
0990 1
0991 1
0992 1
0993 1
0994 1
0995 1
0996 1
0997 1
0998 1
0999 1
1000 1
1001 1
1002 1
1003 1
1004 1
1005 1
1006 1
1007 1
1008 1
1009 1
1010 1
1011 1
1012 1
1013 1
1014 1
1015 1
1016 1
1017 1
1018 1
1019 1
1020 1
1021 1
1022 1
1023 1
1024 1
1025 1
1026 1
1027 1
1028 1
1029 1
1030 1
1031 1
1032 1
1033 1
1034 1
1035 1
1036 1
1037 1
1038 1
1039 1
1040 1
1041 1
1042 1
1043 1
1044 1
1045 1
1046 1
1047 1
1048 1
1049 1
1050 1
1051 1
1052 1
1053 1
1054 1
1055 1
1056 1
1057 1
1058 1
1059 1
1060 1
1061 1
1062 1
1063 1
1064 1
1065 1
1066 1
1067 1
1068 1
1069 1
1070 1
1071 1
1072 1
1073 1
1074 1
1075 1
1076 1
1077 1
1078 1
1079 1
1080 1
1081 1
1082 1
1083 1
1084 1
1085 1
1086 1
1087 1
1088 1
1089 1
1090 1
1091 1
1092 1
1093 1
1094 1
1095 1
1096 1
1097 1
1098 1
1099 1
1100 1
1101 1
1102 1
1103 1
1104 1
1105 1
1106 1
1107 1
1108 1
1109 1
1110 1
1111 1
1112 1
1113 1
1114 1
1115 1
1116 1
1117 1
1118 1
1119 1
1120 1
1121 1
1122 1
1123 1
1124 1
1125 1
1126 1
1127 1
1128 1
1129 1
1130 1
1131 1
1132 1
1133 1
1134 1
1135 1
1136 1
1137 1
1138 1
1139 1
1140 1
1141 1
1142 1
1143 1
1144 1
1145 1
1146 1
1147 1
1148 1
1149 1
1150 1
1151 1
1152 1
1153 1
1154 1
1155 1
1156 1
1157 1
1158 1
1159 1
1160 1
1161 1
1162 1
1163 1
1164 1
1165 1
1166 1
1167 1
1168 1
1169 1
1170 1
1171 1
1172 1
1173 1
1174 1
1175 1
1176 1
1177 1
1178 1
1179 1
1180 1
1181 1
1182 1
1183 1
1184 1
1185 1
1186 1
1187 1
1188 1
1189 1
1190 1
1191 1
1192 1
1193 1
1194 1
1195 1
1196 1
1197 1
1198 1
1199 1
1200 1
1201 1
1202 1
1203 1
1204 1
1205 1
1206 1
1207 1
1208 1
1209 1
1210 1
1211 1
1212 1
1213 1
1214 1
1215 1
1216 1
1217 1
1218 1
1219 1
1220 1
1221 1
1222 1
1223 1
1224 1
1225 1
1226 1
1227 1
1228 1
1229 1
1230 1
1231 1
1232 1
1233 1
1234 1
1235 1
1236 1
1237 1
1238 1
1239 1
1240 1
1241 1
1242 1
1243 1
1244 1
1245 1
1246 1
1247 1
1248 1
1249 1
1250 1
1251 1
1252 1
1253 1
1254 1
1255 1
1256 1
1257 1
1258 1
1259 1
1260 1
1261 1
1262 1
1263 1
1264 1
1265 1
1266 1
1267 1
1268 1
1269 1
1270 1
1271 1
1272 1
1273 1
1274 1
1275 1
1276 1
1277 1
1278 1
1279 1
1280 1
1281 1
1282 1
1283 1
1284 1
1285 1
1286 1
1287 1
1288 1
1289 1
1290 1
1291 1
1292 1
1293 1
1294 1
1295 1
1296 1
1297 1
1298 1
1299 1
1300 1
1301 1
1302 1
1303 1
1304 1
1305 1
1306 1
1307 1
1308 1
1309 1
1310 1
1311 1
1312 1
1313 1
1314 1
1315 1
1316 1
1317 1
1318 1
1319 1
1320 1
1321 1
1322 1
1323 1
1324 1
1325 1
1326 1
1327 1
1328 1
1329 1
1330 1
1331 1
1332 1
1333 1
1334 1
1335 1
1336 1
1337 1
1338 1
1339 1
1340 1
1341 1
1342 1
1343 1
1344 1
1345 1
1346 1
1347 1
1348 1
1349 1
1350 1
1351 1
1352 1
1353 1
1354 1
1355 1
1356 1
1357 1
1358 1
1359 1
1360 1
1361 1
1362 1
1363 1
1364 1
1365 1
1366 1
1367 1
1368 1
1369 1
1370 1
1371 1
1372 1
1373 1
1374 1
1375 1
1376 1
1377 1
1378 1
1379 1
1380 1
1381 1
1382 1
1383 1
1384 1
1385 1
1386 1
1387 1
1388 1
1389 1
1390 1
1391 1
1392 1
1393 1
1394 1
1395 1
1396 1
1397 1
1398 1
1399 1
1400 1
1401 1
1402 1
1403 1
1404 1
1405 1
1406 1
1407 1
1408 1
1409 1
1410 1
1411 1
1412 1
1413 1
1414 1
1415 1
1416 1
1417 1
1418 1
1419 1
1420 1
1421 1
1422 1
1423 1
1424 1
1425 1
1426 1
1427 1
1428 1
1429 1
1430 1
1431 1
1432 1
1433 1
1434 1
1435 1
1436 1
1437 1
1438 1
1439 1
1440 1
1441 1
1442 1
1443 1
1444 1
1445 1
1446 1
1447 1
1448 1
1449 1
1450 1
1451 1
1452 1
14
```

PATFRE
V04-000

E 9
16-Sep-1984 00:16:31 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:52:32 DISK\$VMSMASTER:[PATCH.SRC]PATFRE.B32;1 Page 4
(2)

: 129
0580 1 REQUIRE 'LIBS:PATMSG.REQ';
0754 1 REQUIRE 'SRC\$:SYSSER.REQ';

PAT
V04

: R

```
: R0786 1 SWITCHES LIST (SOURCE);  
: R0787 1  
: R0788 1 EXTERNAL ROUTINE  
: R0789 1     PAT$fao_out;        ! formats a line and outputs to the terminal  
: R0790 1
```

```
: 131      0836 1
: 132      0837 1 ! EXTERNAL REFERENCES
: 133      0838 1
: 134      0839 1
: 135      0840 1
: 136      0841 1 EXTERNAL
: 137          PATSGL_RST BEGN,
: 138          PATSGL_ERRCODE;
: 139      0844 1
: 140      0845 1 ! EQUATED SYMBOLS
: 141      0846 1
: 142      0847 1
: 143      0848 1 LITERAL
: 144          DEBUGGER = TRUE;
: 145      0849 1 ! COMPILE TIME CONDITIONAL FOR DEBUGGING
: 146      L 0850 1
: 147          %IF DEBUGGER
: 148          %THEN
: 149          LITERAL
: 150              NUM_OF_WRK_PAG=128,
: 151              STORESIZE=T512*NUM_OF_WRK_PAG)/4;
: 152          U 0853 1 ! NUMBER OF WORK AREA PAGES
: 153          U 0854 1 ! GET 65K OF WORK AREA FROM P0 NOT P1
: 154          U 0855 1
: 155          U 0856 1 %ELSE
: 156          U 0857 1 LITERAL
: 157          U 0858 1 ! GET WORK SPACE FROM P1 AREA
: 158          U 0859 1 ! virtl . address of symbol table beginning
: 159          U 0860 1
: 160          U 0861 1
: 161          U 0862 1
: 162          U 0863 1 ! OWN STORAGE
: 163          U 0864 1
: 164          U 0865 1
: 165          L 0866 1 OWN
: 166          L 0867 1
: 167          L 0868 1 %IF DEBUGGER
: 168          L 0869 1 %THEN
: 169          L 0870 1 PATSGL_STORAGE: VECTOR[STORESIZE],
: 170          L 0871 1 ! SET ASIDE WORK AREA IN P0
: 171          L 0872 1 PAT$STORE: REF VECTOR,
: 172          L 0873 1 PAT$FREELIST : VECTOR [2];
: 173          L 0874 1
: 174          P 0875 1 BASED_STRUCTURE (FR. 2.
: 175          P 0876 1     SIZE. 0:    0:    32:    0:
: 176          P 0877 1     NEXT. 1:    0:    32:    0;
: 177      : R
```

```

174 0878 1 GLOBAL ROUTINE PAT$FREEINIT : NOVALUE =
175 0879 1
176 0880 1 ++
177 0881 1 FUNCTIONAL DESCRIPTION:
178 0882 1
179 0883 1 PAT$FREEINIT IS CALLED TO INITIALIZE THE FREE STORAGE
180 0884 1 PACKAGE.
181 0885 1
182 0886 1 CALLING SEQUENCE:
183 0887 1
184 0888 1 PAT$FREEINIT ()
185 0889 1
186 0890 1 INPUTS:
187 0891 1
188 0892 1 none
189 0893 1
190 0894 1 IMPLICIT INPUTS:
191 0895 1
192 0896 1 NONE
193 0897 1
194 0898 1 OUTPUTS:
195 0899 1
196 0900 1 NONE
197 0901 1
198 0902 1 IMPLICIT OUTPUTS:
199 0903 1
200 0904 1 PAT$FREELIST AND THE FIRST ELEMENT OF PAT$STORE ARE INITIALIZED.
201 0905 1
202 0906 1 ROUTINE VALUE:
203 0907 1
204 0908 1 NOVALUE
205 0909 1
206 0910 1 SIDE EFFECTS:
207 0911 1
208 0912 1 NONE
209 0913 1 --
210 0914 1
211 0915 2 BEGIN
212 0916 2
213 0917 2 ++
214 0918 2 THIS ROUTINE HAS TWO VERSIONS, DEPENDING UPON WHETHER IT MUST LEAVE THE
215 0919 2 P1 SPACE FOR THE DEBUGGER TO USE, OR CAN USE IT FOR ITS OWN PURPOSES.
216 0920 2 IF THE DEBUGGER IS TO BE LINKED WITH PATCH, THEN THE COMPILE-TIME VARIABLE "DEBUGGER"
217 0921 2 IS SET TO TRUE, CAUSING PATCH TO ALLOCATE THE WORK SPACE FROM PO SPACE
218 0922 2 INSTEAD OF P1 SPACE. OTHERWISE, "DEBUGGER" IS SET TO FALSE AND PATCH
219 0923 2 USES THE P1 SPACE FOR WORK AREA.
220 0924 2 --
221 0925 2
222 0926 2 ++
223 0927 2 FIRST HANDLE THE CASE WHERE THE DEBUGGER IS NOT TO BE LINKED IN.
224 0928 2 --
225 L 0929 2 %IF NOT DEBUGGER
226 U 0930 2 %THEN
227 U 0931 2
228 U 0932 2 LITERAL
229 U 0933 2 START_TAB      = 0.          ! offset for start of symbol table
230 U 0934 2 END_TAB       = 1;          ! offset for end of symbol table.

```

```
: 231      U 0935 2
: 232      U 0936 2 LOCAL      SYMTAB_DESC : VECTOR [2];           ! descriptor for space to create
: 233      U 0937 2
: 234      U 0938 2
: 235      U 0939 2 SYMTAB_DESC [START_TAB] = SYM_TAB_START;
: 236      U 0940 2 SYMTAB_DESC [END_TAB] = SYM_TAB_END;
: 237      U 0941 2 PAT$GL_ERRCODE = $CRETVA (INADR = SYMTAB_DESC, RETADR = SYMTAB_DESC);
: 238      U 0942 2 IF NOT .PAT$GL_ERRCODE
: 239      U 0943 2 THEN
: 240      U 0944 2     SIGNAL (.PAT$GL_ERRCODE);
: 241      U 0945 2
: 242      U 0946 2 !++
: 243      U 0947 2     The create virtual page system service was successful.
: 244      U 0948 2     Now initialize the first few words of the space, and the
: 245      U 0949 2     freelist values.
: 246      U 0950 2 --
: 247      U 0951 2 PAT$STORE = .SYMTAB_DESC [START_TAB];
: 248      U 0952 2
: 249      U 0953 2 !++
: 250      U 0954 2     NOW HANDLE THE CASE WHERE THE DEBUGGER IS LINKED IN.
: 251      U 0955 2 !--
: 252      U 0956 2
: 253      U 0957 2 XELSE
: 254      U 0958 2 PAT$STORE = PAT$GL_STORAGE;
: 255      U 0959 2 XFI
: 256      U 0960 2
: 257      U 0961 2 FR_SIZE (.PAT$STORE) = STORESIZE;
: 258      U 0962 2 FR_NEXT (.PAT$STORE) = NULL;
: 259      U 0963 2 FR_NEXT (PAT$FREELIST) = .PAT$STORE;
: 260      U 0964 2 FR_SIZE (PAT$FREELIST) = 2;
: 261      U 0965 2
: 262      U 0966 2 !++
: 263      U 0967 2     NOW SET THE ADDRESS OF THE START OF THE RST.
: 264      U 0968 2 !--
: 265      U 0969 2 PAT$GL_RST_BEGN = .PAT$STORE;
: 266      U 0970 2
: 267      U 0971 1 END;
```

```
.TITLE PATFRE
.IDENT \V04-000\

.PSECT _PAT$OWN,NOEXE,2

00000 PAT$GL_STORAGE:
.BLKB 65536

10000 PAT$STORE:
.BLKB 4

10004 PAT$FREELIST:
.BLKB 8

.EXTRN PAT$FAO_OUT, PAT$GL_RST_BEGN
.EXTRN PAT$GL_ERRCODE
.WEAK ACCESS_CHECK

.PSECT _PAT$CODE,NOWRT,2
```

52 00000000:	EF 9E 00002	.ENTRY PAT\$FREEINIT, Save R2 ; 0878
62 00000000:	EF 9F 00009	MOVAB PAT\$STORE, P? ; 0958
50 4000 04	62 C 00010	MOVAB PAT\$GL_STORAGE, PAT\$STORE ; 0961
60	8F 3C 00013	MOVL PAT\$STORE, R0
08 A2	A0 D4 00018	MOVZWL #16384, (R0)
04 A2	50 D0 0001B	CLRL 4(R0) ; 0962
00000000G EF	02 D0 0001F	MOVL R0, PAT\$FREELIST+4 ; 0963
	50 D0 00023	MOVL #2, PAT\$FREELIST ; 0964
	04 0002A	MOVL R0, PAT\$GL_RST_BEGN ; 0969
		RET ; 0971

; Routine Size: 43 bytes. Routine Base: _PAT\$CODE + 0000

```
: 269 0972 1 ROUTINE FREEGET (NEED1) =
270 0973 1
271 0974 1 ++
272 0975 1 FUNCTIONAL DESCRIPTION:
273 0976 1
274 0977 1 PROCEDURE FREEGET IS CALLED WITH ONE INPUT ARGUMENT,
275 0978 1 NEED1, WHICH REPRESENTS THE NUMBER OF WORDS OF FREE
276 0979 1 STORAGE NEEDED. IT SEARCHES THE FREELIST FOR A
277 0980 1 NODE THAT HAS AT LEAST THE REQUIRED NUMBER OF WORDS.
278 0981 1 IF THE NODE HAS MORE THAN THE REQUIRED NUMBER OF WORDS,
279 0982 1 THE NODE IS SPLIT INTO TWO NODES, AND ONE IS RETURNED.
280 0983 1
281 0984 1 CALLING SEQUENCE:
282 0985 1
283 0986 1 FREEGET ()
284 0987 1
285 0988 1 INPUTS:
286 0989 1
287 0990 1 NEED1 - THE NUMBER OF WORDS NEEDED OF FREE STORAGE
288 0991 1
289 0992 1 IMPLICIT INPUTS:
290 0993 1
291 0994 1 THE CURRENT STATE OF PATSFREELIST.
292 0995 1
293 0996 1 OUTPUTS:
294 0997 1
295 0998 1 NEED1 - THE ADDRESS OF THE FREE STORAGE ACQUIRED.
296 0999 1
297 1000 1 IMPLICIT OUTPUTS:
298 1001 1
299 1002 1 THE STATE OF PATSFREELIST AND THE COUNT OF WORDS
300 1003 1 OF STORAGE LEFT ARE CHANGED.
301 1004 1
302 1005 1 ROUTINE VALUE:
303 1006 1
304 1007 1 NEED1 - THE ADDRESS OF THE BLOCK
305 1008 1
306 1009 1 SIDE EFFECTS:
307 1010 1
308 1011 1 IF THERE IS NO FREE STORAGE, THEN A SEVERE ERROR IS REPORTED WHICH
309 1012 1 CAUSES AN "UNWIND" FOR THE NEXT COMMAND (INTERACTIVE MODE) OR AN
310 1013 1 EXIT FROM PATCH (BATCH MODE).
311 1014 1 --
312 1015 1
313 1016 2 BEGIN
314 1017 2
315 1018 2 LOCAL
316 1019 2 NEED,
317 1020 2 OLDDNODE,
318 1021 2 LIST,
319 1022 2 HAVE;
320 1023 2
321 1024 2 NEED = .NEED1 + 1 ; ! ONE WORD BIAS FOR SIZE
322 1025 2 IF .NEED LSS 0
323 1026 2 THEN
324 1027 2 SIGNAL (PAT$NDFREE);
325 1028 2 LIST = .FR_NEXT (PAT$FREELIST);
```

```

: 326 1029 2 OLDDNODE = PAT$FREELIST;
: 327 1030 2 WHILE .LIST NEQ NULL
: 328 1031 2 DO
: 329 1032 3 BEGIN
: 330 1033 4 IF ((HAVE = .FR_SIZE (.LIST)) EQ .NEED) ! FIND FIRST FIT
: 331 1034 4 OR (.have GEQ (.need + 2))
: 332 1035 3 THEN
: 333 1036 4 BEGIN
: 334 1037 4 IF .HAVE EQ .NEED ! FOUND
: 335 1038 4 THEN FR_NEXT (.OLDDNODE) = .FR_NEXT (.LIST)
: 336 1039 4 ELSE
: 337 1040 5 BEGIN ! LARGER THAN NEEDED
: 338 1041 5 FR_NEXT (.LIST + .NEED * 4) = .FR_NEXT (.LIST);
: 339 1042 5 FR_SIZE (.LIST + .NEED * 4) = .FR_SIZE (.LIST) - .NEED;
: 340 1043 5 FR_NEXT (.OLDDNODE) = .LIST + .NEED * 4;
: 341 1044 4 END; ! LARGER THAN NEEDED
: 342 1045 4 FR_SIZE (.LIST) = .NEED;
: 343 1046 4
: 344 1047 4 RETURN (.LIST + 4); ! CONTAINS SIZE
: 345 1048 4 END ! FOUND
: 346 1049 3 ELSE LIST = .FR_NEXT (.LIST);
: 347 1050 3 OLDDNODE = .FR_NEXT (.OLDDNODE);
: 348 1051 2 END; ! FIND FIRST FIT
: 349 1052 2 SIGNAL (PAT$_NOFREE); ! Signal error
: 350 1053 2 RETURN NULL
: 351 1054 1 END;

```

			003C 00000 FREEGET: .WORD	Save R2, R3, R4, R5	0972
52	04	AC 0000000G	00 9E 00002 MOVAB LIB\$SIGNAL, R5	#1, NEED1, NEED	1024
			01 C1 00009 ADDL3 1\$		1025
		006D8112	09 18 0000E BGEQ #7176466		1027
			01 FB 00010 PUSHL #1, LIB\$SIGNAL		
	65	51 00000000:	EF DD 00019 CALLS PA\$FREELIST+4, LIST		1028
		53 00000000:	EF 9E 00020 MOVAB PAT\$FREELIST, OLDDNODE		1029
			51 D5 00027 TSTL LIST		1030
			40 13 00029 BEQL 7\$		
		54	61 D0 0002B MOVL (LIST), HAVE		1033
		52	54 D1 0002E CMPL HAVE, NEED		
			09 13 00031 BEQL 3\$		
	50	02	A2 9E 00033 MOVAB 2(R2), R0		1034
		50	54 D1 00037 CMPL HAVE, R0		
			25 19 0003A BLSS 6\$		
		52	54 D1 0003C 3\$: CMPL HAVE, NEED		1037
			07 12 0003F BNEQ 4\$		
	04	A3 04	A1 D0 00041 MOVL 4(LIST), 4(OLDDNODE)		1038
			11 11 00046 BRB 5\$		
60	04	50 6142	DE 00048 4\$: MOVAL (LIST)[NEED], R0		1041
	04	A0	A1 D0 0004C MOVL 4(LIST), 4(R0)		
	61	52	C3 00051 SUBL3 NEED, (LIST), (R0)		1042
	04	A3	50 D0 00055 MOVL R0, 4(OLDDNODE)		1043
	61	52	D0 00059 5\$: MOVL NEED, (LIST)		1045
	50	04	A1 9E 0005C MOVAB 4(LIST), R0		1047

51	04	A1	04 00060	RET				
53	04	A3	00 00061 6\$:	MOVL	4(LIST), LIST			1049
			00 00065	MOVL	4(OLDNODE), OLDNODE			1050
		BC	11 00069	BRB	2\$			1030
65	006D8112	8F	00 0006B 7\$:	PUSHL	#7176466			1052
		01	FB 00071	CALLS	#1, LIB\$SIGNAL			1053
		50	D4 00074	CLRL	R0			1054
		04	00076	RET				

: Routine Size: 119 bytes, Routine Base: _PAT\$CODE + 0028

```
: 353 1055 1 GLOBAL ROUTINE PAT$FREERELEASE (INODE1, NUM1) : NOVALUE =
354 1056 1
355 1057 1 !++
356 1058 1 ! FUNCTIONAL DESCRIPTION:
357 1059 1
358 1060 1 PROCEDURE PAT$FREERELEASE IS CALLED TO RETURN STORAGE
359 1061 1 BACK INTO THE FREE STORAGE POOL. IT MAINTAINS A
360 1062 1 FREELIST IN ASCENDING ORDER OF STORAGE ADDRESSES,
361 1063 1 AND FINDS THE PROPER LOCATION FOR INODE IN THIS
362 1064 1 LIST. THEN IT ATTEMPTS TO MERGE WITH THE LEFT HAND
363 1065 1 NEIGHBOR AND WITH THE RIGHT HAND NEIGHBOR.
364 1066 1
365 1067 1 CALLING SEQUENCE:
366 1068 1
367 1069 1     PAT$FREERELEASE ()
368 1070 1
369 1071 1 INPUTS:
370 1072 1
371 1073 1     INODE1 - ADDRESS OF A NODE
372 1074 1     NUM1   - LENGTH OF A NODE
373 1075 1
374 1076 1 IMPLICIT INPUTS:
375 1077 1
376 1078 1     THE FREE STORAGE POOL, AND THE CURRENT CONTENTS OF PAT$FREELIST.
377 1079 1
378 1080 1 OUTPUTS:
379 1081 1
380 1082 1     NONE
381 1083 1
382 1084 1 IMPLICIT OUTPUTS:
383 1085 1
384 1086 1     AN ERROR MESSAGE CALL ON ERROR
385 1087 1
386 1088 1 ROUTINE VALUE:
387 1089 1
388 1090 1     NOVALUE
389 1091 1
390 1092 1 SIDE EFFECTS:
391 1093 1
392 1094 1     THE STORAGE IS RETURNED TO THE POOL.
393 1095 1 --
394 1096 1
395 1097 2 BEGIN
396 1098 2
397 1099 2 LOCAL
398 1100 2     INODE,
399 1101 2     NUM,
400 1102 2     NODE,
401 1103 2     OLDDNODE;
402 1104 2
403 1105 2     INODE = .INODE1 - 4 ;          ! BIAS FOR SIZE WORD
404 1106 2     NUM = .NUM1 + 1 ;          ! INVISIBLE TO USER
405 1107 2     IF .INODE GEQA .PAT$STORE
406 1108 2         AND .INODE LEQA .PAT$STORE + (STORESIZE * 4) - 1
407 1109 2         AND .FR_SIZE (.INODE) EQL .NUM          ! CORRECT SIZE
408 1110 2     THEN
409 1111 3         BEGIN          ! IN RANGE
```

```

410      1112 3     NODE = .FR_NEXT (PAT$FREELIST);
411      1113 3     OLDDNODE = PAT$FREELIST;
412      1114 3     WHILE .INODE GTRA .NODE AND .NODE NEQ NULL
413      1115 3     DO
414          1116 4     BEGIN
415          1117 4     OLDDNODE = .NODE;
416          1118 4     NODE = .FR_NEXT (.NODE);
417          1119 3     END;
418          1120 3     FR_NEXT (.INODE) = .NODE;
419          1121 3     FR_NEXT (.OLDDNODE) = .INODE;
420          1122 3     INCR I FROM 1 TO 2 DO
421          1123 3     IF .OLDDNODE + .FR_SIZE (.OLDDNODE) * 4 EQLA .FR_NEXT (.OLDDNODE)
422          1124 3     AND .OLDDNODE NEQA PAT$FREELIST !NOT FIRST ON LIST
423          1125 3     THEN
424              1126 4     BEGIN
425              1127 4     FR_SIZE (.OLDDNODE) = .FR_SIZE (.OLDDNODE) + .FR_SIZE (.FR_NEXT (.OLDDNODE));
426              1128 4     FR_NEXT (.OLDDNODE) = .FR_NEXT (.FR_NEXT (.OLDDNODE));
427              1129 4     END
428          1130 3     ELSE OLDDNODE = .FR_NEXT (.OLDDNODE);
429          1131 3     END
430          1132 2     ELSE
431          1133 3     BEGIN
432          1134 3     SIGNAL (IF .FR_SIZE (.INODE) NEQ .NUM
433          1135 3             THEN PATS_FREESIZE
434          1136 3             ELSE PATS_FREERANGE); ! alarm
435          1137 2     END;
436          1138 1     END;

```

					.ENTRY	PAT\$FREERELEASE, Save R2,R3	1055
52	04	53 00000000'	EF 9E 00002		MOVAB	PAT\$STORE R3	1105
51	08	AC	04 C3 00009		SUBL3	#4, INODE1, INODE	1106
	AC	63	01 C1 0000E		ADDL3	#1, NUM1, NUM	1107
		52	D1 00013		CMPL	INODE, PAT\$STORE	
		65	1F 00016		BLSSU	6\$	
50		63 0000FFFF	8F C1 00018		ADDL3	#65535, PAT\$STORE, R0	1108
		50	52 D1 00020		CMPL	INODE, R0	
		58	1A 00023		BGTRU	6\$	
		51	62 D1 00025		CMPL	(INODE), NUM	1109
		53	12 00028		BNEQ	6\$	
		50	08 A3 D0 0002A		MOVL	PAT\$FREELIST+4, NODE	1112
		51	04 A3 9E 0002E		MOVAB	PAT\$FREELIST, OLDDNODE	1113
		50	52 D1 00032	1\$:	CMPL	INODE, NODE	1114
			0D 1B 00035		BLEQU	2\$	
			50 D5 00037		TSTL	NODE	
			09 13 00039		BEQL	2\$	
		51	50 D0 0003B		MOVL	NODE, OLDDNODE	1117
		50	04 A0 D0 0003E		MOVL	4(NODE), NODE	1118
			EE 11 00042		BRB	1\$	1114
04	A2	50	D0 00044	2\$:	MOVL	NODE, 4(INODE)	1120
04	A1	52	D0 00048		MOVL	INODE, 4(OLDDNODE)	1121
52		01	D0 0004C		MOVL	#1, I	1122
50		61	D0 0004F	3\$:	MOVL	(OLDDNODE), R0	1123
50		6140	DE 00052		MOVAL	(OLDDNODE)[R0], R0	

	04 A1	50 D1 00056	CMPL R0, 4(OLDNODE)	:
	50 04	18 12 0005A	BNEQ 4\$	
	50 04	A3 9E 0005C	MOVAB PAT\$FREELIST, R0	1124
	50	51 D1 00060	CMPL OLDNODE, R0	
		0F 13 00063	BEQL 4\$	
	61 04	B1 C0 00065	ADDL2 @4(OLDNODE), (OLDNODE)	1127
	50 04	A1 D0 00069	MOVL 4(OLDNODE), R0	1128
D3	04 A1	04 A0 D0 0006D	MOVL 4(R0), 4(OLDNODE)	
		04 11 00072	BRB 5\$	1123
	51 04	A1 D0 00074 4\$:	MOVL 4(OLDNODE), OLDNODE	1130
	52	02 F3 00078 5\$:	AOBLEQ #2, I, 3\$	1123
		04 0007C	RET	1107
	51	62 D1 0007D 6\$:	CMPL (INODE), NUM	1134
		08 13 00080	BEQL 7\$	
	006D80BA	8F DD 00082	PUSHL #7176378	
		06 11 00088	BRB 8\$	
	006D80B2	8F DD 0008A 7\$:	PUSHL #7176370	
	00000000G 00	01 FB 00090 8\$:	CALLS #1, LIB\$SIGNAL	
		04 00097	RET	1138

: Routine Size: 152 bytes, Routine Base: _PAT\$CODE + 00A2

```

: 438
: 439 1139 1 GLOBAL ROUTINE PAT$FREEZ (NEED) =
: 440 1140 1
: 441 1141 1 +++
: 442 1142 1 FUNCTIONAL DESCRIPTION:
: 443 1143 1 CALLS FREEGET TO ALLOCATE STORAGE, AND CLEARS IT
: 444 1144 1
: 445 1145 1 CALLING SEQUENCE:
: 446 1146 1
: 447 1147 1 PAT$FREEZ ()
: 448 1148 1
: 449 1149 1
: 450 1150 1 INPUTS:
: 451 1151 1 LENGTH OF AREA WANTED (IN LONGWORDS)
: 452 1152 1
: 453 1153 1 IMPLICIT INPUTS:
: 454 1154 1
: 455 1155 1 NONE
: 456 1156 1
: 457 1157 1 OUTPUTS:
: 458 1158 1
: 459 1159 1 ADDRESS OF AREA
: 460 1160 1
: 461 1161 1 IMPLICIT OUTPUTS:
: 462 1162 1
: 463 1163 1 NONE
: 464 1164 1
: 465 1165 1
: 466 1166 1 ROUTINE VALUE:
: 467 1167 1
: 468 1168 1 NOVALUE
: 469 1169 1
: 470 1170 1 SIDE EFFECTS:
: 471 1171 1
: 472 1172 1 NONE
: 473 1173 1 --
: 474 1174 1 BEGIN
: 475 1175 2
: 476 1176 2 LOCAL
: 477 1177 2 P:
: 478 1178 2
: 479 1179 2 IF (P = FREEGET (.NEED)) NEQ 0
: 480 1180 2 THEN ZEROCOR (.P, .NEED);
: 481 1181 2 RETURN .P
: 482 1182 1
: 483 1183 1 END:

```

				04	007C 00000	.ENTRY	PAT\$FREEZ, Save R2,R3,R4,R5,R6	1139
		FEE7	CF		01 DD 00002	PUSHL	NEED	1180
			56		01 FB 00005	CALLS	#1, FREEGET	
					50 D0 0000A	MOVL	R0, P	
					0B 13 0000D	BEQL	1\$	
	50	00	04	AC	02 78 0000F	ASHL	#2, NEED, R0	
50				6E	00 2C 00014	MOVCS	#0, (SP), #0, R0, (P)	1181

PATFRE
V04-000

E 10
16-Sep-1984 00:16:31
14-Sep-1984 12:52:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[PATCH.SRC]PATFRE.B32;1

Page 17
(6)

PA1
VO4

50 66 00019
56 D0 0001A 1\$:
04 0001D MOVL P, R0
RET

; 1182
; 1183

: Routine Size: 30 bytes. Routine Base: _PATSCODE + 013A

```
484 1184 1 GLOBAL ROUTINE PAT$REPORT_FREE =  
485 1185 1  
486 1186 1 ++  
487 1187 1 FUNCTIONAL DESCRIPTION:  
488 1188 1 REPORTS THE NUMBER OF BYTES LEFT IN FREE STORAGE.  
489 1189 1  
490 1190 1  
491 1191 1 CALLING CONVENTION:  
492 1192 1  
493 1193 1 PAT$REPORT_FREE ()  
494 1194 1  
495 1195 1 INPUTS:  
496 1196 1  
497 1197 1 NONE  
498 1198 1  
499 1199 1 IMPLICIT INPUTS:  
500 1200 1  
501 1201 1 THE ELEMENTS OF THE FREE LIST.  
502 1202 1  
503 1203 1 OUTPUTS:  
504 1204 1  
505 1205 1 NONE  
506 1206 1  
507 1207 1 IMPLICIT OUTPUTS:  
508 1208 1  
509 1209 1 NONE  
510 1210 1  
511 1211 1 ROUTINE VALUE:  
512 1212 1  
513 1213 1 THE NUMBER OF BYTES OF STORAGE THAT IS FREE.  
514 1214 1  
515 1215 1 SIDE EFFECTS:  
516 1216 1  
517 1217 1 NONE  
518 1218 1 --  
519 1219 1  
520 1220 2 BEGIN  
521 1221 2  
522 1222 2 LOCAL  
523 1223 2 COUNT,  
524 1224 2 POINTER;  
525 1225 2  
526 1226 2 COUNT = 0;  
527 1227 2  
528 1228 2 ++  
529 1229 2 STEP THROUGH THE FREE LIST.  
530 1230 2 !--  
531 1231 2 POINTER = .FR_NEXT (PAT$FREELIST);  
532 1232 2 WHILE .POINTER NEQ NULL  
533 1233 2 DO  
534 1234 3 BEGIN  
535 1235 3 COUNT = .COUNT + .FR_SIZE (.POINTER);  
536 1236 3 POINTER = .FR_NEXT (-.POINTER);  
537 1237 2 END;  
538 1238 2 RETURN .COUNT * 4  
539 1239 1 END;
```

50

54

	0000 00000	.ENTRY PAT\$REPORT_FREE, Save nothing	: 1184
51 00000000'	50 D4 00002	CLRL COUNT	: 1226
	EF D0 00004	MOVL PAT\$FREELIST+4, POINTER	: 1231
	09 13 0000B	BEQL 2\$: 1232
50	61 C0 0000D	ADDL2 (POINTER), COUNT	: 1235
51	04 A1 D0 00010	MOVL 4(POINTER), POINTER	: 1236
	F5 11 00014	BRB 1\$: 1232
50	04 C4 00016	MULL2 #4, R0	: 1238
	04 00019	RET	: 1239

; Routine Size: 26 bytes. Routine Base: _PAT\$CODE + 0158

: 541 1240 1 END
: 542 1241 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
PAT\$OWN	65548	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
PAT\$CODE	370	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----			Pages Mapped	Processing Time
	Total	Loaded	Percent		
\$_255\$DUA28:[SYSLIB]STARLET.L32;1	9776	4	0	581	00:01.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/VARIANT:1/LIS=LIS\$:PATFRE/OBJ=OBJ\$:PATFRE MSRC\$:PATFRE/UPDATE=(ENH\$:PATFRE)

Size: 370 code + 65548 data bytes
Run Time: 00:18.7
Elapsed Time: 00:53.2
Lines/CPU Min: 3979
Lexemes/CPU-Min: 48086
Memory Used: 117 pages
Compilation Complete

0301 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PATERR
LIS

PATEXA
LIS

PATIHD
LIS

PATINS
LIS

PATINT
LIS